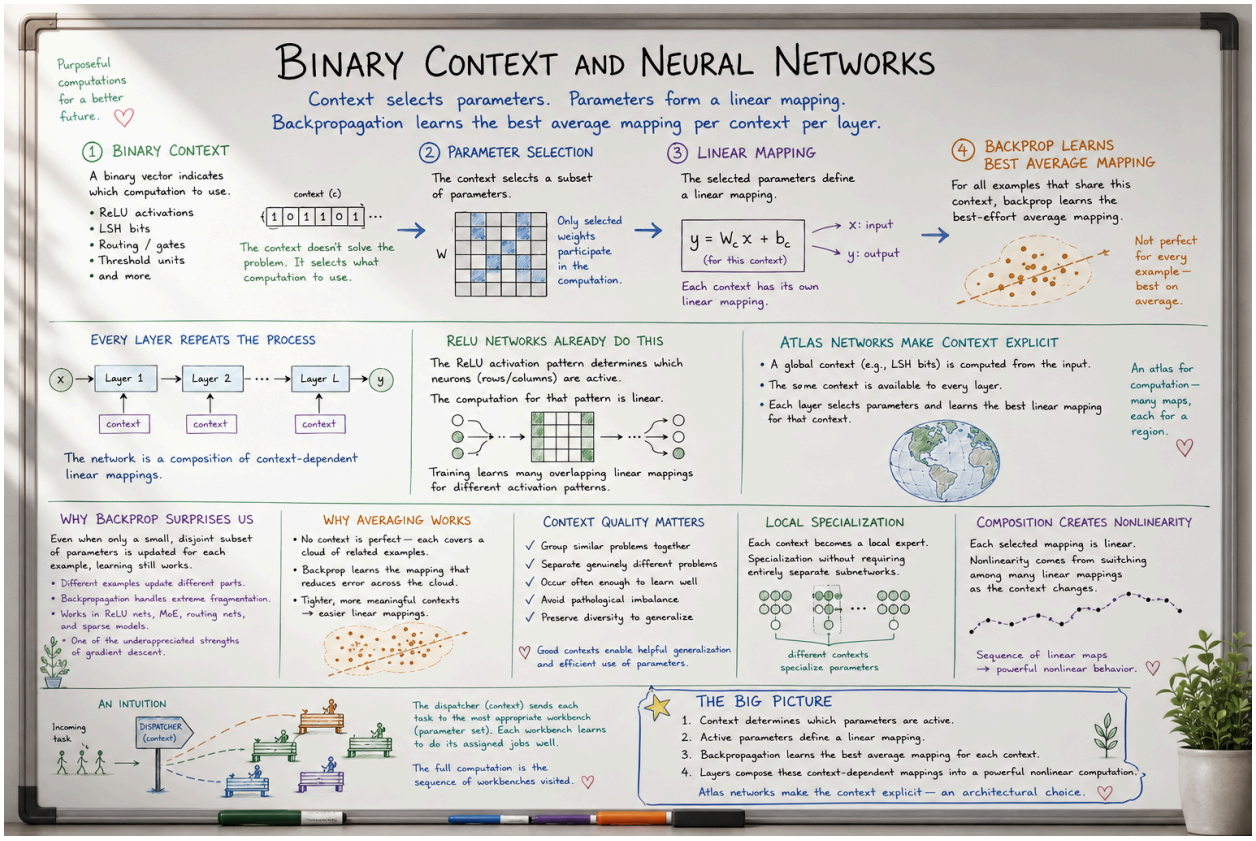
Purposeful computations for a better future.
BINARY CONTEXT AND NEURAL NETWORKS
Context selects parameters. Parameters form a linear mapping.
Backpropagation learns the best average mapping per context per layer.
1 BINARY CONTEXT
A binary vector indicates which computation to use.
• ReLU activations
• LSH bits
• Routing / gates
• Threshold units
• and more
context (c)
1 0 1 1 0 1 ...
The context doesn't solve the problem. It selects what computation to use.
2 PARAMETER SELECTION
The context selects a subset of parameters.
W
Only selected weights participate in the computation.
3 LINEAR MAPPING
The selected parameters define a linear mapping.
y = W_c x + b_c
(for this context)
x: input
y: output
Each context has its own linear mapping.
4 BACKPROP LEARNS BEST AVERAGE MAPPING
For all examples that share this context, backprop learns the best-effort average mapping.
Not perfect for every example — best on average.
EVERY LAYER REPEATS THE PROCESS
x
Layer 1
Layer 2
Layer L
y
context
context
context
The network is a composition of context-dependent linear mappings.
RELU NETWORKS ALREADY DO THIS
The ReLU activation pattern determines which neurons (rows/columns) are active.
The computation for that pattern is linear.
Training learns many overlapping linear mappings for different activation patterns.
ATLAS NETWORKS MAKE CONTEXT EXPLICIT
• A global context (e.g., LSH bits) is computed from the input.
• The same context is available to every layer.
• Each layer selects parameters and learns the best linear mapping for that context.
An atlas for computation — many maps, each for a region.
WHY BACKPROP SURPRISES US
Even when only a small, disjoint subset of parameters is updated for each example, learning still works.
• Different examples update different parts.
• Backpropagation handles extreme fragmentation.
• Works in ReLU nets, MoE, routing nets, and sparse models.
• One of the underappreciated strengths of gradient descent.
WHY AVERAGING WORKS
• No context is perfect — each covers a cloud of related examples.
• Backprop learns the mapping that reduces error across the cloud.
• Tighter, more meaningful contexts → easier linear mappings.
CONTEXT QUALITY MATTERS
✓ Group similar problems together
✓ Separate genuinely different problems
✓ Occur often enough to learn well
✓ Avoid pathological imbalance
✓ Preserve diversity to generalize
Good contexts enable helpful generalization and efficient use of parameters.
LOCAL SPECIALIZATION
Each context becomes a local expert.
Specialization without requiring entirely separate subnetworks.
different contexts specialize parameters
COMPOSITION CREATES NONLINEARITY
Each selected mapping is linear.
Nonlinearity comes from switching among many linear mappings as the context changes.
Sequence of linear maps → powerful nonlinear behavior.
AN INTUITION
Incoming task
DISPATCHER (context)
The dispatcher (context) sends each task to the most appropriate workbench (parameter set). Each workbench learns to do its assigned jobs well.
The full computation is the sequence of workbenches visited.
THE BIG PICTURE
1. Context determines which parameters are active.
2. Active parameters define a linear mapping.
3. Backpropagation learns the best average mapping for each context.
4. Layers compose these context-dependent mappings into a powerful nonlinear computation.
Atlas networks make the context explicit — an architectural choice.

A practical way to think about neural networks.